

Learning Entropy Production via Neural Networks

Dong-Kyum Kim^{1,*}, Youngkyoung Bae^{1,*}, Sangyun Lee¹, and Hawoong Jeong^{1,2,†}

¹*Department of Physics, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea*

²*Center for Complex Systems, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea*

(Received 13 March 2020; revised 12 June 2020; accepted 11 September 2020; published 2 October 2020)

This Letter presents a neural estimator for entropy production (NEEP), that estimates entropy production (EP) from trajectories of relevant variables without detailed information on the system dynamics. For steady state, we rigorously prove that the estimator, which can be built up from different choices of deep neural networks, provides stochastic EP by optimizing the objective function proposed here. We verify the NEEP with the stochastic processes of the bead spring and discrete flashing ratchet models and also demonstrate that our method is applicable to high-dimensional data and can provide coarse-grained EP for Markov systems with unobservable states.

DOI: 10.1103/PhysRevLett.125.140604

Nonequilibrium states are ubiquitously observed from colloidal particles to biological systems [1–6]. Injection of energy, lack of relaxation time, or broken detailed balance are ordinary sources of nonequilibrium, and in general, such systems are in contact with a heat bath such as a fluid. Thus, to describe the behavior of a nonequilibrium system, it is necessary to investigate the energetics of the system; however, experimentally, heat flow is difficult to measure directly [7–10]. In this case, measuring the entropy production (EP) can be one remedy to estimate heat flow in a nonequilibrium system [10–12].

Many techniques have been developed to accurately measure EP, such as approaches calculating probability currents and density [9,13]. These methods require detailed information from a governing equation though, so to address this issue, a few methods to estimate the EP rate without such detailed information have been proposed, including the plug-in method [14–16], the compression-based estimator [15–19], and the thermodynamic uncertainty relation (TUR)-based estimator [13,20–23]. The plug-in and compression-based methods estimate the EP rate through the Kullback-Leibler divergence, but they are only applicable for discrete state variables. And, while the TUR-based approach has recently been adopted in frameworks for the exact estimation of EP rates and distributions in short time limits [21–23], estimating stochastic EP remains an unsolved issue for continuous state variables.

Various fields in physics have been employing machine learning (ML) to solve a wide range of nontrivial problems, such as identifying relevant variables [24–26], identifying phase transitions [27–32], quantum many-body problems [33–40], and others [41]. Likewise, ML has also been applied to EP rate estimation [23,42], as well as classification of the direction of time’s arrow [43]. Relatedly, in the ML community, a recent work by Rahaman *et al.* [44] proposed a neural network to measure an entropylike

quantity by unsupervised learning; however, the quantity was not physically well defined, i.e., it had no scale. To the best of our knowledge, estimating EP using neural networks has yet to be explored.

In this Letter, we propose the neural estimator for entropy production (NEEP), which can estimate stochastic EP from the time-series data of relevant variables without detailed information on the dynamics of the system. For Markov chain trajectory s_1, s_2, \dots, s_L , we build a function h_θ that takes two states, s_t and s_{t+1} , where θ denotes the trainable neural network parameters. As shown in Fig. 1(a), the output of NEEP is defined as

$$\Delta S_\theta(s_t, s_{t+1}) \equiv h_\theta(s_t, s_{t+1}) - h_\theta(s_{t+1}, s_t). \quad (1)$$

Here, $\Delta S_\theta(s_t, s_{t+1})$ satisfies the antisymmetric relation $\Delta S_\theta(s_t, s_{t+1}) = -\Delta S_\theta(s_{t+1}, s_t)$. We define the objective function to be maximized as

$$J(\theta) = \mathbb{E}_t \mathbb{E}_{s_t \rightarrow s_{t+1}} [\Delta S_\theta(s_t, s_{t+1}) - e^{-\Delta S_\theta(s_t, s_{t+1})}], \quad (2)$$

where \mathbb{E}_t denotes the expectation over t , which is uniformly sampled from $\{1, \dots, L-1\}$, and $\mathbb{E}_{s_t \rightarrow s_{t+1}}$ is the expectation

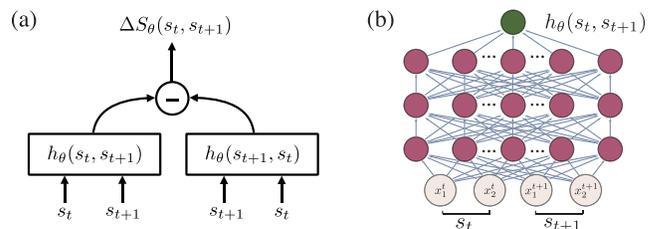


FIG. 1. (a) Architecture of the NEEP. (b) Illustration of a MLP with three hidden layers for an $N = 2$ bead-spring model where $s_t = (x_1^t, x_2^t)$.

over transition $s_t \rightarrow s_{t+1}$. If detailed balance is satisfied, then the transition $s \rightarrow s'$ and its reverse transition $s' \rightarrow s$ equally appear in the ensemble of the trajectories. In this case, the optimized ΔS_θ is zero for all possible transitions, but if detailed balance is broken, then ΔS_θ becomes larger due to more irreversible transitions. In steady state, $J(\theta)$ can be written as

$$J[h] = \sum_{i,j} p_i T_{ij} [(h_{ij} - h_{ji}) - e^{-(h_{ij}-h_{ji})}], \quad (3)$$

where we set $h_{ij} \equiv h(s_i, s_j)$, $p_i \equiv p(s_i)$ is the steady-state probability density, and $T_{ji} \equiv p(s_i, t+1|s_j, t)$ is a propagator. Because the neural networks tune output $h_{\alpha\beta} \equiv h(s_\alpha, s_\beta)$ by optimizing θ , the maximum condition for Eq. (3) becomes

$$\begin{aligned} 0 &= \partial_{h_{\alpha\beta}} J[h] \\ &= \sum_{i,j} [p_i T_{ij} (1 + e^{-(h_{ij}-h_{ji})}) (\delta_{i\alpha} \delta_{j\beta} - \delta_{i\beta} \delta_{j\alpha})] \\ &= p_\alpha T_{\alpha\beta} (1 + e^{-(h_{\alpha\beta}-h_{\beta\alpha})}) - p_\beta T_{\beta\alpha} (1 + e^{-(h_{\beta\alpha}-h_{\alpha\beta})}). \end{aligned} \quad (4)$$

Then the solution for the optimization problem is

$$h_{\alpha\beta} - h_{\beta\alpha} = -\ln(p_\beta T_{\beta\alpha} / p_\alpha T_{\alpha\beta}), \quad (5)$$

which is the definition of stochastic entropy production [12] when $T_{ji} = \tilde{T}_{ij}$. Here, \tilde{T}_{ij} is the time-reversal propagator of T_{ij} . This proof supports the ability of our NEEP to learn appropriate EP. We maximize Eq. (2) via the stochastic gradient ascent method that is widely used in deep learning literature [45,46]. See the Supplemental Material [47] for the training and evaluation details.

To validate our approach, we estimate the EP of two widely studied nonequilibrium systems: the bead-spring model for continuous state variables [5,13,42,51] and the discrete flashing ratchet model for discrete state variables [15,16,52]. To attempt more challenging problems, we additionally apply NEEP to high-dimensional continuous models and a hidden Markov model.

In the bead-spring model, N beads are coupled to the nearest beads or boundary walls by springs and contacted with thermal heat baths at different temperatures, as described in Fig. 2(a). For displacements x_1, x_2, \dots, x_N , the dynamics of N beads is governed by an overdamped Langevin equation

$$\dot{x}_i(\tau) = A_{ij} x_j(\tau) + \sqrt{2k_B T_i / \gamma} \xi_i(\tau), \quad (6)$$

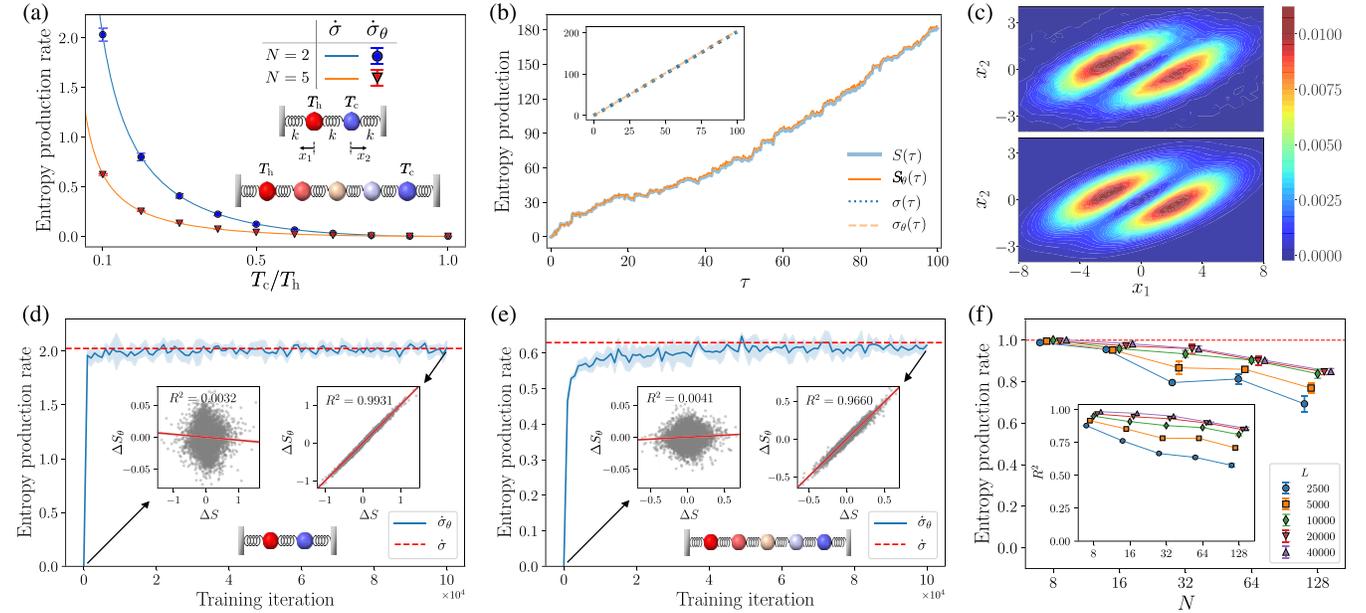


FIG. 2. (a) Entropy production rate as a function of T_c/T_h for models with two and five beads. The solid lines (symbols) indicate the analytical EP rate $\dot{\sigma}$ (estimated EP rate $\hat{\sigma}_\theta$). (b) Cumulative EP over time τ along a single trajectory, which is randomly sampled from the test set. The inset shows the ensemble-averaged EP. (c) Local EP rate as a function of x_1 and x_2 . Top, bottom: the NEEP and analytical results, respectively. (d), (e) $\hat{\sigma}_\theta$ with respect to training iteration for (d) two beads and (e) five beads. The left (right) inset corresponds to results before (after) training, showing scatter plots between ΔS and ΔS_θ with a fitted linear regression line (solid red line). The results in (b)–(e) are performed at $T_c/T_h = 0.1$. (f) Results of NEEP for high-dimensional bead-spring models. $\hat{\sigma}_\theta$ as a function of N for each number of steps (L) are plotted with five different markers, as indicated in the legend. The R^2 values of the linear regression between ΔS and ΔS_θ are shown in the inset. The red dashed line denotes $\dot{\sigma}$. Error bars and shaded areas represent the standard deviation of estimations from five independently trained estimators.

where $A_{ij} = (-2\delta_{i,j} + \delta_{i,j+1} + \delta_{i+1,j})k/\gamma$. Here, k is a spring constant, γ is the Stokes friction coefficient, and the temperature T_i of each heat bath linearly varies from T_h to T_c . ξ_i is an independent Gaussian white noise satisfying $\mathbb{E}[\xi_i(\tau)\xi_j(\tau')] = \delta_{ij}\delta(\tau - \tau')$, where \mathbb{E} denotes the ensemble average. We set all the parameters to be dimensionless and $k_B = k = \gamma = 1$. The linearly varying temperature induces a thermodynamic force that drives the system to a non-equilibrium state.

To attempt EP estimation in a system with continuous variables, we first consider $N = 2$ and $N = 5$ bead-spring models. Here, $\dot{\sigma}$ is the analytical value of the ensemble-averaged EP rate [47]; Fig. 2(a) plots $\dot{\sigma}$ for $N = 2$ (5) with a blue (orange) solid line. As illustrated in Fig. 1(b), we employ a three-hidden-layer multilayer perceptron (MLP) for h_θ . See the Supplemental Material [47] for the configuration and robustness of the architecture. For training and test sets, we numerically sampled 10^3 positional trajectories in steady state for each model. Each trajectory was sampled with time step $\Delta\tau = 10^{-2}$ [53], and the total number of steps L is 10^4 . We present the training results at $T_c/T_h = 0.1$ in Figs. 2(b)–2(e). Note that all reported results in Fig. 2 are from the test set. We also demonstrate the estimation ability of NEEP with various L [47].

For the $N = 2$ case, as shown in Fig. 2(b), it is observed that our estimator provides accurate values not only for the ensemble average but also for a single trajectory over τ . Here, $S(\tau) \equiv \sum_{i=0}^{\tau/\Delta\tau} \Delta S(s_i, s_{i+1})$ and $\sigma(\tau) \equiv \mathbb{E}[S(\tau)]$, where ΔS is the analytic stochastic EP per $\Delta\tau$. Figure 2(c) shows that the local EP rate over the displacement space (x_1, x_2) calculated by NEEP (top panel) is the same as the analytical solution (bottom panel). The local EP rate from NEEP at (x_1, x_2) is measured by averaging the EP rate produced when a particle passes through the point (x_1, x_2) .

To check the training process, we plot the estimated values of $\dot{\sigma}_\theta$ over training iteration in Fig. 2(d). The dashed red line indicates $\dot{\sigma}$. Insets in Fig. 2(d) are scatter plots between ΔS_θ and ΔS in a randomly sampled single trajectory. As can be seen in the left inset, there is no correlation between ΔS_θ and ΔS before training. But after training (right inset), ΔS_θ is well fitted to ΔS (coefficient of determination $R^2 = 0.9931$).

We apply the same process to the $N = 5$ bead-spring model, where estimating ΔS and $\dot{\sigma}$ using the thermodynamic force is difficult due to the curse of dimensionality [13]. The result shows that ΔS_θ is again well fitted to ΔS with $R^2 = 0.9660$ [see Fig. 2(e)]. We also train our estimator at T_c in the range of 1–10 with $T_h = 10$, as indicated in Fig. 2(a), and verify that NEEP provides the exact EP rate with small errors. Notably, these results are from the test set, implying that NEEP can be generalized to estimate EP even for unseen data.

Estimating EP in high-dimensional Langevin systems has not been explored because of the curse of

dimensionality [22]. While a recent work [23] has made estimations of EP rates up to $N = 15$ using TUR, here, we apply NEEP to bead-spring models with $N = 8, 16, 32, 64$, and 128. For each N , we set $T_h = 10$ and T_c to a value where $\dot{\sigma} = 1$ [47]. By increasing the training data points ($10^3 L$), we can see that $\dot{\sigma}_\theta$ for each N approaches one in Fig. 2(f). Although EP rate estimation errors of over 10% are seen for $N = 64$ and 128, the R^2 values support that NEEP was able to learn the stochastic EP with appreciable correlations [see the inset in Fig. 2(f)]. Note that, with an increasing number of beads, the architecture of NEEP does not change except for the number of input nodes ($2N$), which means that our neural estimator's computation time and the number of parameters are linearly proportional to N . Based on these points, we show that NEEP can efficiently mitigate the curse of dimensionality through a neural network.

Next, we demonstrate our method on the discrete flashing ratchet model [52], which consists of a particle moving in a one-dimensional periodic lattice. The particle is in contact with a heat bath at temperature T and drifts in a periodic asymmetric sawtooth potential [see Fig. 3(b)]. For brevity, we set $k_B = T = 1$. In this model, the particle state has two variables, x and η , where $x \in \{0, 1, 2\}$ is the position and $\eta \in \{\text{ON}, \text{OFF}\}$ is the on-off potential; the state is indicated as $i \equiv (i, \text{ON})$ and $i' \equiv (i, \text{OFF})$. Transition rates between each state $s \in \{0, 1, 2, 0', 1', 2'\}$ are defined as $k_{ij} = e^{(V_j - V_i)/2}$ and $k_{i'j'} = 1$ for $i \neq j$, where V_i is the potential at i that switches on and off at rate $r = 1$, i.e., $k_{ii'} = k_{i'i} = r$. As in a previous work [15], we generate a series of states and remove the information of the times

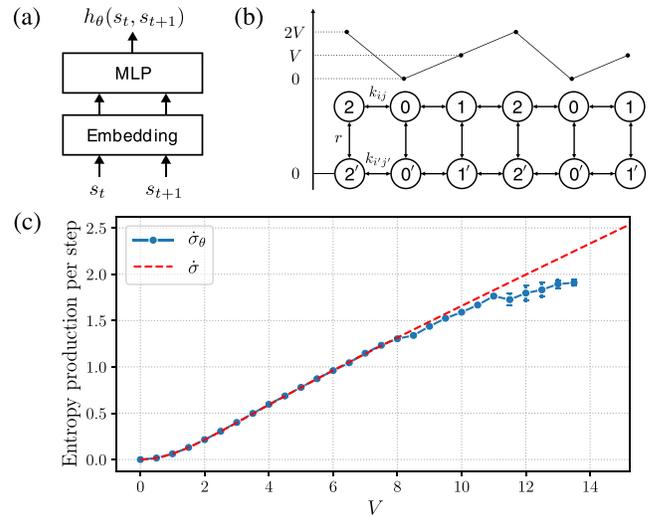


FIG. 3. (a) NEEP architecture for discrete state Markov chains. (b) Schematic of a discrete flashing ratchet model. (c) Entropy production per step as a function of potential V . Error bars represent the standard deviation of $\dot{\sigma}_\theta$ from five independently trained estimators.

when transitions occur; in this case, the analytic EP per step is given as $\dot{\sigma} = \sum_{\alpha,\beta} p(\alpha,\beta)(V_\alpha - V_\beta)$.

We construct the NEEP as shown in Fig. 3(a) using an embedding layer that transforms a discrete state into a trainable continuous vector called an embedding vector. After the transformation, we feed the two embedding vectors of states s_t and s_{t+1} to the MLP [47]. From a set of different potential values, we sampled two single trajectories with $L = 10^6$ steps for each potential V ($0 \leq V \leq 15$); one trajectory is used for training and the other for testing. For the training data, we build five NEEPs, randomly initialized with five different random seeds for each potential. Figure 3(c) shows that $\dot{\sigma}$ is within the error bar of the NEEP estimations of EP per step $\dot{\sigma}_\theta$, where $V \leq 8$. For V in the range 8–14, the overfitting [46] problem occurs due to a lack of transitions from low to high potential, which leads to an underestimation of $\dot{\sigma}_\theta$ [Fig. 3(c)]. See the Supplemental Material [47] for a more detailed discussion on how we address the overfitting issue. For $14 \leq V$, the probability to detect the $0 \rightarrow 2$ transition is below 0.5 in our simulation with $L = 10^6$. In this case, $\dot{\sigma}_\theta$ diverges because of no observation of the $0 \rightarrow 2$ transition (see Fig. S5 in the Supplemental Material [47]).

So far, Markovian systems with completely observable states have been tested; however, full state information cannot often be accessed, with only some coarse-grained variables typically available. In such cases, the EP of a coarse-grained trajectory, called coarse-grained EP, is measurable [54–57]. To test for coarse-grained EP estimation, we assume that the on-off information η is now inaccessible [15,16]. To address this problem, we build h_θ with a recurrent neural network (RNN), a popular network to consider memory effects in time-series data. We employ a gated recurrent unit (GRU) [58] for the RNN. As shown in Fig. 4(a), the RNN version of NEEP (RNEEP) takes input as a series of states with a sequence length of n , and the outputs of the GRU are averaged over the sequence and then fed to a single layer feed forward neural network, which is the last layer. Now, the RNEEP output is defined as $\Delta S_\theta(\mathbf{x}_t^n) \equiv h_\theta(\mathbf{x}_t^n) - h_\theta(\tilde{\mathbf{x}}_t^n)$, and the objective function is defined as

$$J(\theta) = \mathbb{E}_t \mathbb{E}_{(\mathbf{x}_t^n, \boldsymbol{\eta}_t^n)} [\Delta S_\theta(\mathbf{x}_t^n) - e^{-\Delta S_\theta(\mathbf{x}_t^n)}], \quad (7)$$

where

$$\mathbf{x}_t^n = (x_t, x_{t+1}, \dots, x_{t+n-1}), \quad \boldsymbol{\eta}_t^n = (\eta_t, \eta_{t+1}, \dots, \eta_{t+n-1}).$$

Here, $\tilde{\mathbf{x}}_t^n$ is the time-reversed trajectory of \mathbf{x}_t^n . In steady state, the solution for this optimization problem is the stochastic coarse-grained EP along the trajectory \mathbf{x}^n (see the Supplemental Material [47] for the proof),

$$\Delta S_\theta(\mathbf{x}^n) = -\ln \frac{\sum_{\tilde{\boldsymbol{\eta}}^n} p(\tilde{\mathbf{x}}^n, \tilde{\boldsymbol{\eta}}^n)}{\sum_{\boldsymbol{\eta}^n} p(\mathbf{x}^n, \boldsymbol{\eta}^n)} = -\ln \frac{p(\tilde{\mathbf{x}}^n)}{p(\mathbf{x}^n)}. \quad (8)$$

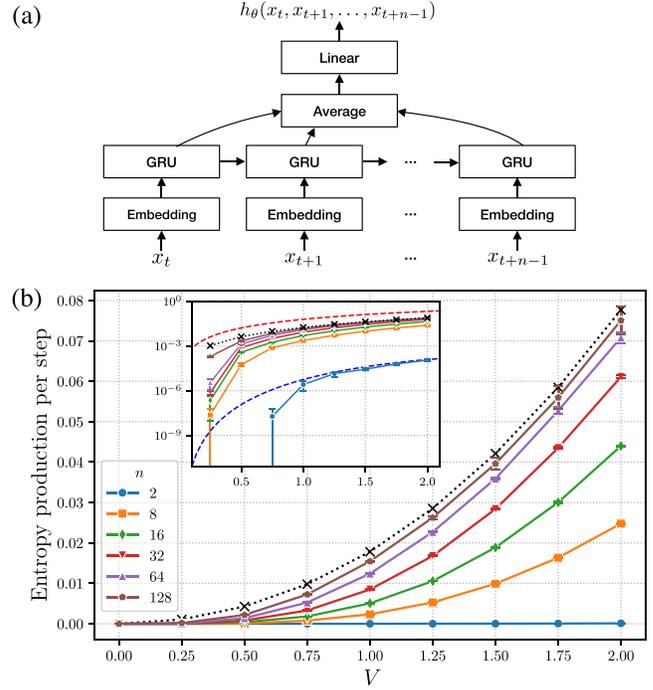


FIG. 4. (a) RNEEP architecture for a hidden Markov model. (b) Results of RNEEP for a partial information problem. The estimations $\dot{\Sigma}_\theta^n$ as a function of potential V for each sequence length n are plotted with six different markers as shown in the legend. The black \times 's are the semianalytical values of $\dot{\Sigma}^\infty$. The inset shows a plot of the y axis in log scale. The red (blue) dashed line denotes the analytic value of $\dot{\sigma}$ ($\dot{\Sigma}^2$). See Fig. S6 in the Supplemental Material [47] for a comparison with $\dot{\sigma}$ in linear scale. Error bars represent the standard deviation of estimations from five independently trained estimators.

Here, the ensemble-averaged coarse-grained EP of trajectory \mathbf{x}^n per step is denoted as $\dot{\Sigma}^n \equiv \mathbb{E}[-\ln(p(\tilde{\mathbf{x}}^n)/p(\mathbf{x}^n))]/(n-1)$. In general, $\dot{\Sigma}^n$ provides a lower bound on the actual EP per step $\dot{\sigma}$ [15,16].

For $0 \leq V \leq 2$, we train the RNEEP with six different sequence lengths, $n = 2, 8, 16, 32, 64$, and 128 , for maximizing Eq. (7) using the position trajectory with $L = 5 \times 10^7$. As can be seen in Fig. 4(b), with increasing sequence length n , the estimation of RNEEP ($\dot{\Sigma}_\theta^n$) approaches the semianalytical value of the coarse-grained EP per step for $n \rightarrow \infty$ ($\dot{\Sigma}^\infty$) [16]. We can verify that $\dot{\Sigma}_\theta^2$ is well fitted to the analytic value of $\dot{\Sigma}^2$, but it remains difficult to estimate for $V \leq 1$ [see the inset in Fig. 4(b)], because the number of transitions between any two positions, e.g., $x \rightarrow y$ or $y \rightarrow x$, appears almost equally in the trajectory. While directly estimating Eq. (8) by counting the frequency of \mathbf{x}^n is not possible for $n \geq 16$ due to the curse of dimensionality, the RNEEP can resolve this issue and enable us to estimate the coarse-grained EP up to $n = 128$ (see Figs. S7 and S8 in the Supplemental Material [47]).

In previous approaches [10], estimation of the probability distribution and the probability current was essential

to quantify how far the system is out of equilibrium. As NEEP does not require such estimation or detailed information of the system, we expect our estimator to be applicable to various fields such as active matter, biological systems, information machines, electronic devices, and others. This approach will be particularly useful to investigate the stochastic energetics and spatiotemporal patterns of dissipated energy in various systems. We further expect our method to be applicable to the understanding of complex nonequilibrium systems, e.g., soft biological assemblies [51] or molecular motors with hidden internal states [59]. As a future work, modifying our NEEP method to estimate EP in more general nonequilibrium systems like time-dependent states will be intriguing.

The code for NEEP, implemented in PyTorch [60], is available in Ref. [61].

This study was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) (Grant No. NRF-2017R1A2B3006930).

*These authors equally contributed to this work.

[†]hjeong@kaist.edu

- [1] P. Martin, A. J. Hudspeth, and F. Jülicher, Comparison of a hair bundle's spontaneous oscillations with its response to mechanical stimulation reveals the underlying active process, *Proc. Natl. Acad. Sci. U.S.A.* **98**, 14380 (2001).
- [2] E. Ben-Isaac, Y. K. Park, G. Popescu, F. L. H. Brown, N. S. Gov, and Y. Shokef, Effective Temperature of Red-Blood-Cell Membrane Fluctuations, *Phys. Rev. Lett.* **106**, 238103 (2011).
- [3] S. C. Weber, A. J. Spakowitz, and J. A. Theriot, Nonthermal ATP-dependent fluctuations contribute to the in vivo motion of chromosomal loci, *Proc. Natl. Acad. Sci. U.S.A.* **109**, 7338 (2012).
- [4] C. Battle, C. M. Ott, D. T. Burnette, J. Lippincott-Schwartz, and C. F. Schmidt, Intracellular and extracellular forces drive primary cilia movement, *Proc. Natl. Acad. Sci. U.S.A.* **112**, 1410 (2015).
- [5] C. Battle, C. P. Broedersz, N. Fakhri, V. F. Geyer, J. Howard, C. F. Schmidt, and F. C. MacKintosh, Broken detailed balance at mesoscopic scales in active biological systems, *Science* **352**, 604 (2016).
- [6] J. Gladrow, N. Fakhri, F. C. MacKintosh, C. F. Schmidt, and C. P. Broedersz, Broken Detailed Balance of Filament Dynamics in Active Networks, *Phys. Rev. Lett.* **116**, 248301 (2016).
- [7] T. Harada and S.-i. Sasa, Equality Connecting Energy Dissipation with a Violation of the Fluctuation-Response Relation, *Phys. Rev. Lett.* **95**, 130602 (2005).
- [8] S. Toyabe, H.-R. Jiang, T. Nakamura, Y. Murayama, and M. Sano, Experimental test of a new equality: Measuring heat dissipation in an optically driven colloidal system, *Phys. Rev. E* **75**, 011122 (2007).
- [9] B. Lander, J. Mehl, V. Blickle, C. Bechinger, and U. Seifert, Noninvasive measurement of dissipation in colloidal systems, *Phys. Rev. E* **86**, 030401(R) (2012).
- [10] F. S. Gnesotto, F. Mura, J. Gladrow, and C. P. Broedersz, Broken detailed balance and non-equilibrium dynamics in living systems: A review, *Rep. Prog. Phys.* **81**, 066601 (2018).
- [11] K. Sekimoto, *Stochastic Energetics* (Springer-Verlag, Berlin, 2010).
- [12] U. Seifert, Stochastic thermodynamics, fluctuation theorems and molecular machines, *Rep. Prog. Phys.* **75**, 126001 (2012).
- [13] J. Li, J. M. Horowitz, T. R. Gingrich, and N. Fakhri, Quantifying dissipation using fluctuating currents, *Nat. Commun.* **10**, 1666 (2019).
- [14] Q. Wang, S. R. Kulkarni, and S. Verdú, Divergence estimation of continuous distributions based on data-dependent partitions, *IEEE Trans. Inf. Theory* **51**, 3064 (2005).
- [15] É. Roldán and J. M. R. Parrondo, Estimating Dissipation from Single Stationary Trajectories, *Phys. Rev. Lett.* **105**, 150607 (2010).
- [16] É. Roldán and J. M. R. Parrondo, Entropy production and Kullback-Leibler divergence between stationary trajectories of discrete systems, *Phys. Rev. E* **85**, 031129 (2012).
- [17] J. Ziv and N. Merhav, A measure of relative entropy between individual sequences with application to universal classification, *IEEE Trans. Inf. Theory* **39**, 1270 (1993).
- [18] R. Avinery, M. Kornreich, and R. Beck, Universal and Accessible Entropy Estimation Using a Compression Algorithm, *Phys. Rev. Lett.* **123**, 178102 (2019).
- [19] S. Martiniani, P. M. Chaikin, and D. Levine, Quantifying Hidden Order out of Equilibrium, *Phys. Rev. X* **9**, 011031 (2019).
- [20] A. C. Barato and U. Seifert, Thermodynamic Uncertainty Relation for Biomolecular Processes, *Phys. Rev. Lett.* **114**, 158101 (2015).
- [21] S. K. Manikandan, D. Gupta, and S. Krishnamurthy, Inferring Entropy Production from Short Experiments, *Phys. Rev. Lett.* **124**, 120603 (2020).
- [22] T. Van Vu, V. T. Vo, and Y. Hasegawa, Entropy production estimation with optimal current, *Phys. Rev. E* **101**, 042138 (2020).
- [23] S. Otsubo, S. Ito, A. Dechant, and T. Sagawa, Estimating entropy production by machine learning of short-time fluctuating currents, *Phys. Rev. E* **101**, 062106 (2020).
- [24] P. Mehta and D. J. Schwab, An exact mapping between the variational renormalization group and deep learning, [arXiv:1410.3831](https://arxiv.org/abs/1410.3831).
- [25] M. Koch-Janusz and Z. Ringel, Mutual information, neural networks and the renormalization group, *Nat. Phys.* **14**, 578 (2018).
- [26] S.-H. Li and L. Wang, Neural Network Renormalization Group, *Phys. Rev. Lett.* **121**, 260601 (2018).
- [27] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, *Nat. Phys.* **13**, 431 (2017).
- [28] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, *Nat. Phys.* **13**, 435 (2017).
- [29] J. Venderley, V. Khemani, and E.-A. Kim, Machine Learning Out-of-Equilibrium Phases of Matter, *Phys. Rev. Lett.* **120**, 257204 (2018).

- [30] Y.-H. Liu and E. P. L. van Nieuwenburg, Discriminative Cooperative Networks for Detecting Phase Transitions, *Phys. Rev. Lett.* **120**, 176401 (2018).
- [31] M. J. S. Beach, A. Golubeva, and R. G. Melko, Machine learning vortices at the Kosterlitz-Thouless transition, *Phys. Rev. B* **97**, 045207 (2018).
- [32] P. Zhang, H. Shen, and H. Zhai, Machine Learning Topological Invariants with Neural Networks, *Phys. Rev. Lett.* **120**, 066401 (2018).
- [33] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* **355**, 602 (2017).
- [34] D.-L. Deng, X. Li, and S. Das Sarma, Quantum Entanglement in Neural Network States, *Phys. Rev. X* **7**, 021021 (2017).
- [35] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Machine Learning Phases of Strongly Correlated Fermions, *Phys. Rev. X* **7**, 031038 (2017).
- [36] K. Choo, G. Carleo, N. Regnault, and T. Neupert, Symmetries and Many-Body Excitations with Neural-Network Quantum States, *Phys. Rev. Lett.* **121**, 167204 (2018).
- [37] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, *Nat. Phys.* **14**, 447 (2018).
- [38] M. J. Hartmann and G. Carleo, Neural-Network Approach to Dissipative Quantum Many-Body Dynamics, *Phys. Rev. Lett.* **122**, 250502 (2019).
- [39] A. Nagy and V. Savona, Variational Quantum Monte Carlo Method with a Neural-Network Ansatz for Open Quantum Systems, *Phys. Rev. Lett.* **122**, 250501 (2019).
- [40] F. Vicentini, A. Biella, N. Regnault, and C. Ciuti, Variational Neural-Network Ansatz for Steady States in Open Quantum Systems, *Phys. Rev. Lett.* **122**, 250503 (2019).
- [41] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, *Rev. Mod. Phys.* **91**, 045002 (2019).
- [42] F. S. Gnesotto, G. Gradziuk, P. Ronceray, and C. P. Broedersz, Learning the non-equilibrium dynamics of Brownian movies, [arXiv:2001.08642](https://arxiv.org/abs/2001.08642).
- [43] A. Seif, M. Hafezi, and C. Jarzynski, Machine learning the thermodynamic arrow of time, *Nat. Phys.* (2020).
- [44] N. Rahaman, S. Wolf, A. Goyal, R. Remme, and Y. Bengio, Learning the arrow of time for problems in reinforcement learning, in *International Conference on Learning Representations* (2020), <https://openreview.net/forum?id=rylJkpEtwS>.
- [45] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature (London)* **521**, 436 (2015).
- [46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).
- [47] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevLett.125.140604> for the (1) training setup and algorithm; (2) evaluation method, architecture details, and robustness; (3) analytic description of the bead-spring model; (4) overfitting issue; (5) training details; and (6) proof for coarse-grained entropy production, which includes Refs. [16,46,48–50].
- [48] V. Nair and G. E. Hinton, Rectified linear units improve restricted Boltzmann machines, in *Proceedings of the 27th International Conference on Machine Learning* (Omni-press, Haifa, 2010), pp. 807–814.
- [49] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [50] U. Seifert, Entropy Production along a Stochastic Trajectory and an Integral Fluctuation Theorem, *Phys. Rev. Lett.* **95**, 040602 (2005).
- [51] F. Mura, G. Gradziuk, and C. P. Broedersz, Nonequilibrium Scaling Behavior in Driven Soft Biological Assemblies, *Phys. Rev. Lett.* **121**, 038002 (2018).
- [52] A. Ajdari and J. Prost, Mouvement induit par un potentiel périodique de basse symétrie: diélectrophorese pulsée, *C. R. Acad. Sci. Paris II* **315**, 1635 (1992).
- [53] The time interval between the current and next states can affect the estimation of NEEP. For large enough time intervals, the correlation between the two states weakens and NEEP may provide imprecise estimations.
- [54] A. Gomez-Marín, J. M. R. Parrondo, and C. Van den Broeck, Lower bounds on dissipation upon coarse graining, *Phys. Rev. E* **78**, 011107 (2008).
- [55] M. Esposito, Stochastic thermodynamics under coarse graining, *Phys. Rev. E* **85**, 041125 (2012).
- [56] M. Polettoni and M. Esposito, Effective Thermodynamics for a Marginal Observer, *Phys. Rev. Lett.* **119**, 240601 (2017).
- [57] L. Dabelow, S. Bo, and R. Eichhorn, Irreversibility in Active Matter Systems: Fluctuation Theorem and Mutual Information, *Phys. Rev. X* **9**, 021009 (2019).
- [58] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Doha, Qatar, 2014), pp. 1724–1734.
- [59] I. A. Martínez, G. Bisker, J. M. Horowitz, and J. M. R. Parrondo, Inferring broken detailed balance in the absence of observable currents, *Nat. Commun.* **10**, 3542 (2019).
- [60] A. Paszke *et al.*, PyTorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc., Vancouver, 2019), pp. 8024–8035.
- [61] D.-K. Kim, Y. Bae, S. Lee, and H. Jeong, kdkyum/neep: Neural Estimator for Entropy Production (2020), Version: v1.0.0, <https://doi.org/10.5281/zenodo.4041300>.